

## V. Claims

We claim:

1. A statistical record purge procedure comprising the improvement of ensuring there is adequate temporary memory for storing new statistical records before the procedure initiates deletion of older statistical records in permanent memory, the new statistical records comprising statistical records generated while the older statistical records are being deleted.
2. The procedure of claim 1 wherein the adequate temporary memory comprises ninety percent or more free memory space.
3. The procedure of claim 1 comprising the further improvement of waiting a predetermined period of time if there is inadequate temporary memory, and then repeating the procedure of claim 1.
4. The procedure of claim 1 wherein the predetermined period of time comprises approximately thirty seconds.
5. The procedure of claim 1 wherein the temporary memory space comprises a log space.
6. The procedure of claim 1 wherein the permanent memory utilizes indices or pointers and the indices or pointers are reset following the deletion of older statistical records.
7. The procedure of claim 1 wherein the deletion of older statistical records occurs less frequently than hourly but at least once daily.
8. A statistical record purge procedure comprising the improvement of scheduling deletion of older statistical records during a period when relatively few new statistical records are generated.

9. The procedure of claim 8 wherein the permanent memory utilizes indices or pointers and the indices or pointers are reset following the deletion of older statistical records.
10. The procedure of claim 8 wherein the deletion of older statistical records occurs less frequently than hourly but at least once daily.
11. An improved method of purging statistical records from a temporary memory in a service provider network, the improvement comprising:
  - (A) ensuring there is adequate temporary memory for storing new statistical records before the procedure initiates deletion of older statistical records in permanent memory, the new statistical records comprising statistical records generated while the older statistical records are being deleted.
12. The method of claim 11 further comprising:
  - (B) waiting a predetermined period of time if there is inadequate temporary memory; and
  - (C) repeating step (A).
13. The method of claim 12 wherein the predetermined period comprises approximately thirty seconds.
14. The method of claim 11 further comprising:
  - (D) updating any indices or pointers utilized by the permanent memory following the deletion of older statistical records in the permanent memory.

## APPENDIX A

### check\_purge.sh

---

```

prog=`basename $0` 

if test $# -lt 1
then
    echo "Need dbname" >>/opt/BulkStats/etc/$prog.log
    exit 1
fi

if test -s /opt/BulkStats/etc/$prog.log
then
    dte=`date +%d%b%Y`
    mv -f /opt/BulkStats/etc/$prog.log \
        /opt2/BulkStats.var/$prog.log@$dte
    compress -f /opt2/BulkStats.var/$prog.log@$dte
fi

DBNAME=NAVIS-STATN
export prog DBNAME

if ping -I 1 navis-statn 24 1|grep "0 packets received"
then
    echo "navis-statn not responding at `date`" \
        >>/opt/BulkStats/etc/$prog.log 2>&1
    rm -f /BulkStats/data/NXStatisticsCbxGbx.purging
    exit 1
fi

#####
# lock out other db type cron jobs !
#####
touch /BulkStats/data/NXStatisticsCbxGbx.purging

#####
# this is a routine to check for an empty db log, if not
# sleep up to 10 minutes waiting for one
#####
check_db ()
{
#####
# loop up to 12 times, i.e. 6 minutes, until the logfile is
# close to 100% free
#####

cnt=12
while true
do
    remsh $DBNAME -l sybase -e /opt/sybase/query >/tmp/$prog.## 2>&1 <<!
sp_helpdb $1
go
quit
exit
!
    LogSize=`cat /tmp/$prog.## | grep _log | awk '{print $2,$6}'`
    rm -f /tmp/$prog.##

```

```

Size=`echo $LogSize | awk '{print $1}' | cut -f1 -d.`
Free=`echo $LogSize | awk '{print $2}'`

Size=`expr $Size \* 1024000`
Free=`expr $Free \* 100000`

WFree=`expr $Free \V $Size`
RFree=`expr $Free \% $Size`
RFree=`echo $RFree | cut -c1-2` 

echo "$1 has $WFree.$RFree free logspace at `date`" \
>>/opt/BulkStats/etc/$prog.log 2>&1

if test $WFree -gt 85
then
    echo >>/opt/BulkStats/etc/$prog.log 2>&1
    break
else
    sleep 30
fi

cnt=`expr $cnt - 1`
if test $cnt -lt 0
then
    echo "$prog: aborting because of full db log for $1" \
    >>/opt/BulkStats/etc/$prog.log 2>&1
fi
done
}

export MinTime MaxTime
echo "$prog:\tStarting at `date`\n" >>/opt/BulkStats/etc/$prog.log

check_db "$1"

for x in TrkStat CktStat TrunkStat FrCktStat FrLportStat ATMCKtStat ATMPrtStat \
        ATMSvcStat ATMTrkStat ATMLPrtNiStat ATMLPrtTrkStat ATMFirstTrkStat \
        ATMOptTrkStat IpLportStat SmdsLportStat
do

    MinTime=`remsh $DBNAME -l sybase -e /opt/sybase/query <<!
use $1
go
select min(startTime) from $x
go
quit
exit
!` 

    if echo $MinTime | egrep "NULL|Msg" >/dev/null
    then
        echo "No table data for $x\n" >>/opt/BulkStats/etc/$prog.log
        continue
    fi

    MinTime=`echo $MinTime | awk '{print $2}'` 
    echo "$x:\t$MinTime <$MinTime>" >>/opt/BulkStats/etc/$prog.log

    if test $MinTime -le 950000000
    then

```

```

        echo "$x:\tbad number for MinTime" \
              >>/opt/BulkStats/etc/$prog.log
        continue
    fi

    MaxTime=`/BulkStats/bin/perl5 -e '$utcseconds=time();print "$utcseconds\n"'`
    echo "$x:\tMaxTime <$MaxTime>" >>/opt/BulkStats/etc/$prog.log

    if test $MaxTime -le 950000000
    then
        echo "$x:\tbad number for MaxTime" \
              >>/opt/BulkStats/etc/$prog.log
        continue
    fi

    DiffTime=`expr $MaxTime - $MinTime`
    DiffTime=`expr $DiffTime / 86400`

    echo "$x:\tnumber of days in database is $DiffTime\n" \
          >>/opt/BulkStats/etc/$prog.log

    ######
    # delete all records older than 30 days
    #####
    if test $DiffTime -gt 31
    then
        Ttime=`expr $DiffTime - 31`
        DelTime=0
        export DelTime

        while true
        do
            if test $Ttime -eq 0
            then
                break
            fi

            DelTime=`expr "$MinTime" + 86400`
            MinTime=`expr "$MinTime" + 86400`
            export DelTime MinTime

            echo "$x:\tDelTime <$DelTime>" \
                  >>/opt/BulkStats/etc/$prog.log
            echo "$x:\tdelete $x where startTime < $DelTime at `date`\n" \
                  >>/opt/BulkStats/etc/$prog.log

            #####
            # execute the 'query' file on remote server so
            # passwd is not exposed !
            #####
            remsh $DBNAME -I sybase -e /opt/sybase/query \
                  >>/opt/BulkStats/etc/$prog.log 2>&1 <<!
use $1
go
delete $x where startTime < $DelTime
go
checkpoint
go
quit

```

```
exit
!
DiffTime=`expr $DiffTime - 1`
echo "\n$x:\tnumber of days left in database is $DiffTime" \
      >>/opt/BulkStats/etc/$prog.log

Ttime=`expr $Ttime - 1`

echo >>/opt/BulkStats/etc/$prog.log
check_db "$1"
done
fi
done

rm -f /BulkStats/data/NXStatisticsCbxGbx.purging
echo "$prog:\tEnding at `date`\n" >>/opt/BulkStats/etc/$prog.log
```

## APPENDIX B

check\_stats.sh

```
prog='basename $0'

if test $# -lt 1
then
    echo "Need dbname" >>/opt/BulkStats/etc/$prog.log
    exit 1
fi

>/opt/BulkStats/etc/$prog.log

##if test -s /opt/BulkStats/etc/$prog.log
##then
##mv -f /opt/BulkStats/etc/$prog.log \
##    ##/opt/BulkStats/etc/$prog.log.old
##fi

DBNAME=NAVIS-STATN
export prog DBNAME
echo >>/opt/BulkStats/etc/$prog.log

for x in TrkStat CktStat TrunkStat FrCktStat FrLportStat ATMCKtStat ATMPrtStat \
        ATMSvcStat ATMTrkStat ATMLPrtNiStat ATMLPrtTrkStat ATMFirstTrkStat \
        ATMOptTrkStat IpLportStat SmdsLportStat
do
    echo "Starting update statistics $x at `date`" >>/opt/BulkStats/etc/$prog.log
    remsh $DBNAME -l sybase -e /opt/sybase/query \
        >>/opt/BulkStats/etc/$prog.log 2>&1 <<!
use $1
go
update statistics $x
go
quit
exit
!
echo "Ending update statistics $x at `date`\n" \
    >>/opt/BulkStats/etc/$prog.log

done

echo "$prog:\tEnding at `date`\n" >>/opt/BulkStats/etc/$prog.log
```